IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Aurelien Legendre
June 2025

Table of Contents

Executive Summary

SpaceX Falcon 9 first stage Landing Prediction

SpaceX advertises Falcon 9 rocket launches with a cost of 62 million dollars when other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.

Based on multiple factors (launch site, pay load mass, orbit type...), we will **predict if the Falcon 9 first stage will land succesfully.**

Introduction

A rocket first stage launches is a huge company cost, and reusing it can be an important cost saving.

Launch a first stage rocket, companies knows how to do it ; but to reuse it, the reality is not so easy... And depend on multiple factors :

In this report, based on historical data, we will analyze keys success factors and predict the Falcon 9 first stage Landing by answering the questions below :

- What is the **best Orbit ?**
- What is the best **Pay Load mass range ?**
- What is the best **Launch Site ?**

Section 1

Methodology

Methodology

Perform data collection :

- Using Web Scraping Wikipedia
- Using API by GET request ‚BoosterVersion’ + ‚LaunchSite’ + ‚PayLoad’ + ‚CoreData’
- Creation of a dictionnary and then incorporate the dictionnary into a dataframe
- Filtering the data on Falcon 9

Perform data wrangling

- The column LandingPad was containing 26 ‚Nan’ / ‚null’ values replaced by 0 (zero)

Perform exploratory data analysis (EDA) using visualization and SQL

- Analysis of Data using pandas and SQL
- Data Visualization using seaborn by creating differents charts

Perform interactive visual analytics using Folium and Plotly Dash

- Interactive Dashboard Creation showing important information

Perform predictive analysis using classification models

- Machine Leaning models & evaluation

Data Collection – SpaceX API

Source :

GET request from API : **https://api.spacexdata.com**

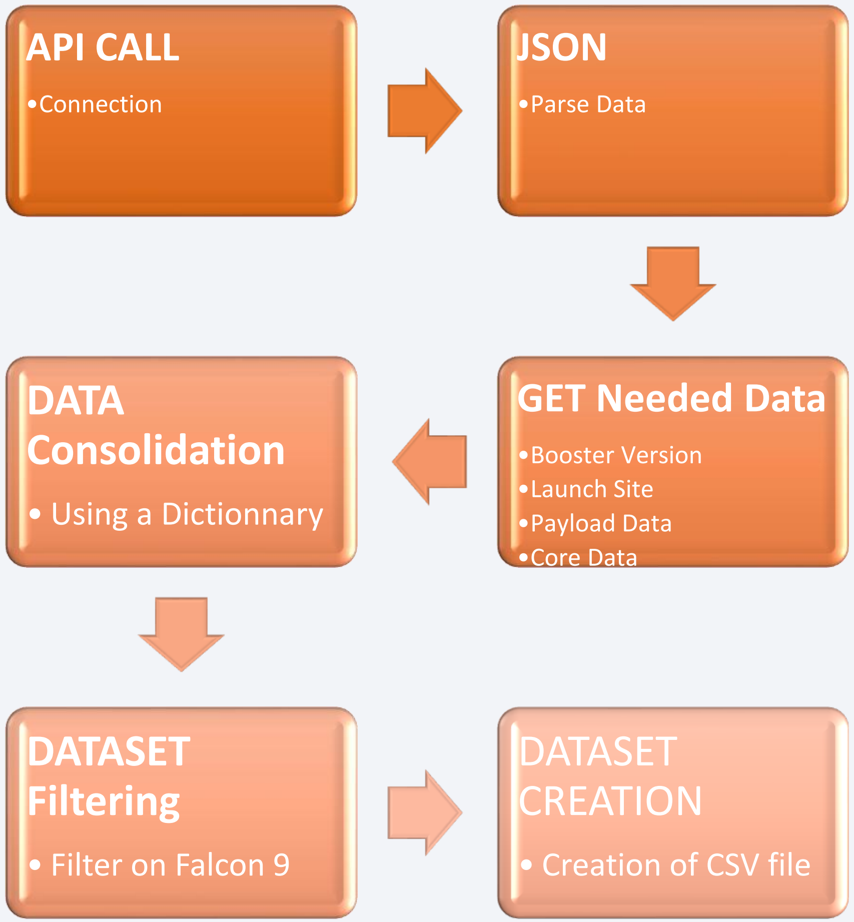
Notebook :

All details on the GitHub link (here)

File : jupyter-labs-spacex-data-collection-api.ipynb

Purpose :

Collect needed data to create a dataset



Data Collection - Scraping

Source :

GET data from WIKIPEDIA :
https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

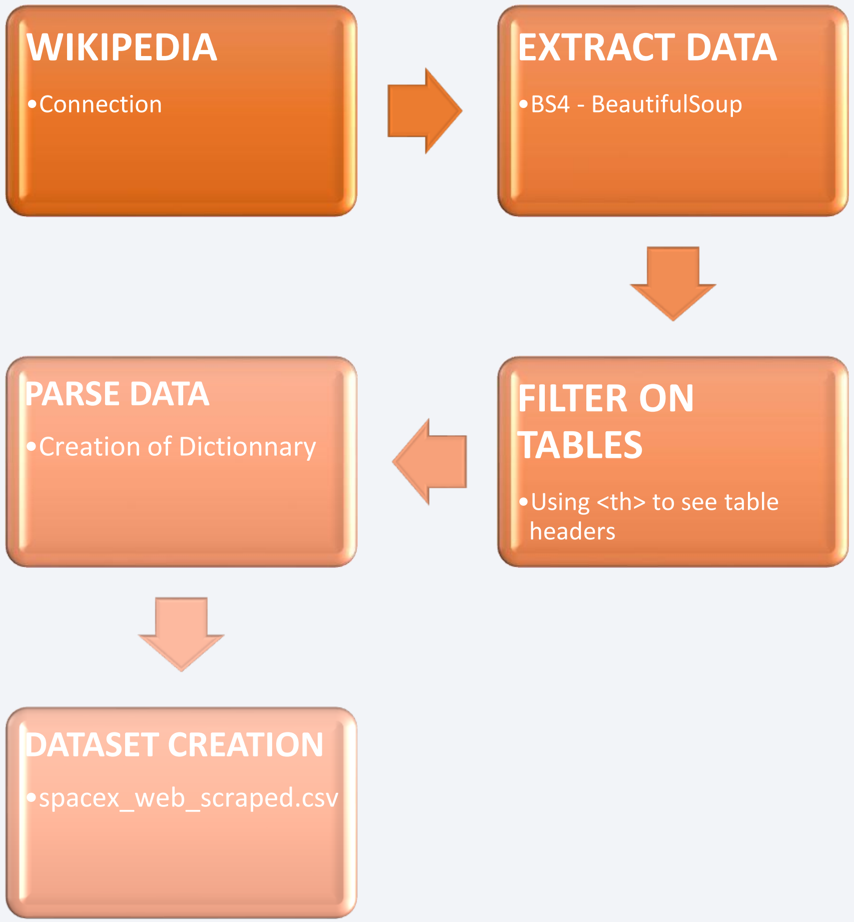
Notebook :

All details on the GitHub link (here)

File : jupyter-labs-webscraping.ipynb

Purpose :

Collect needed data to create a dataset



Data Wrangling

Source :

CSV File : **https dataset_part_1.csv**
Created at the end of the process Data Wrangling

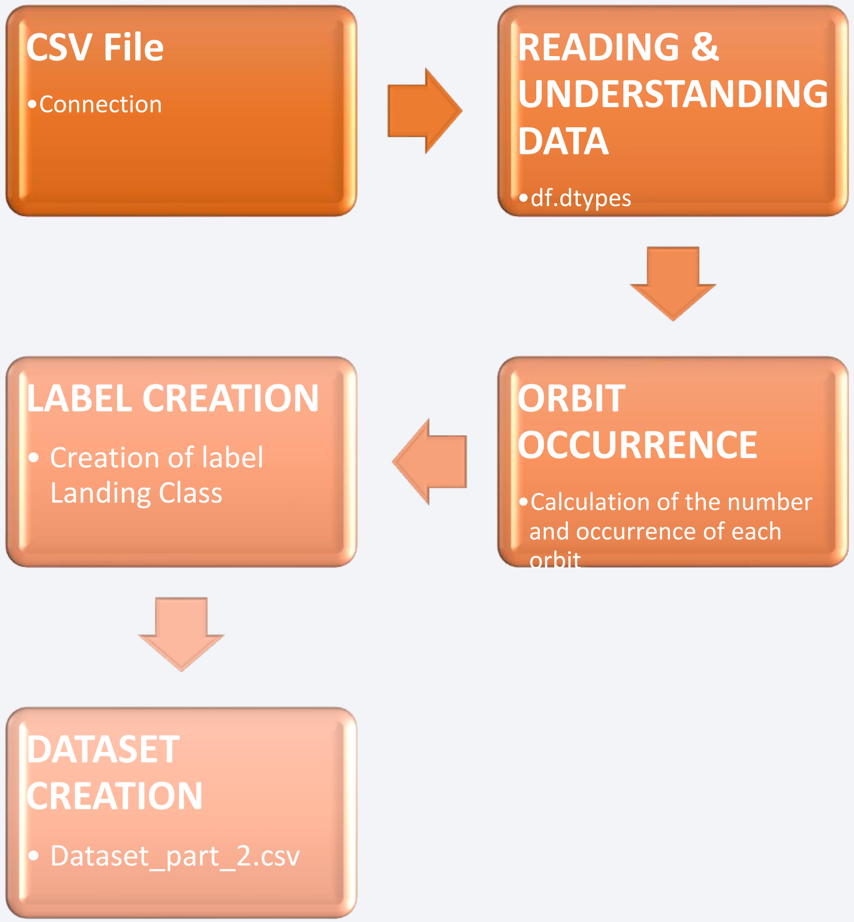
Notebook :

All details on the GitHub link (here)

File : labs-jupyter-spacex-Data wrangling.ipynb

Purpose :

Clean collected data to focus only on useful information before analysis



EDA with Data Visualization

Source :

CSV File : **https dataset_part_2.csv** **Created at the end of the process Data Wrangling**

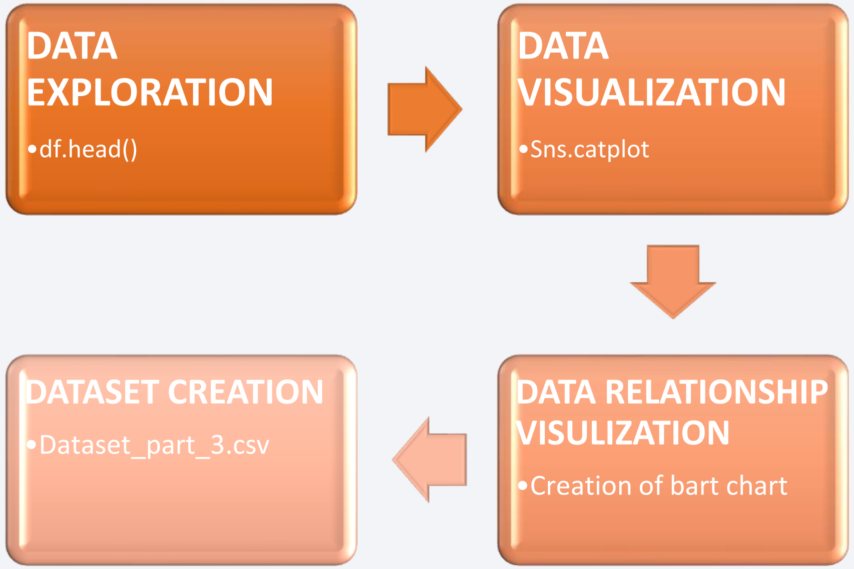
Notebook :

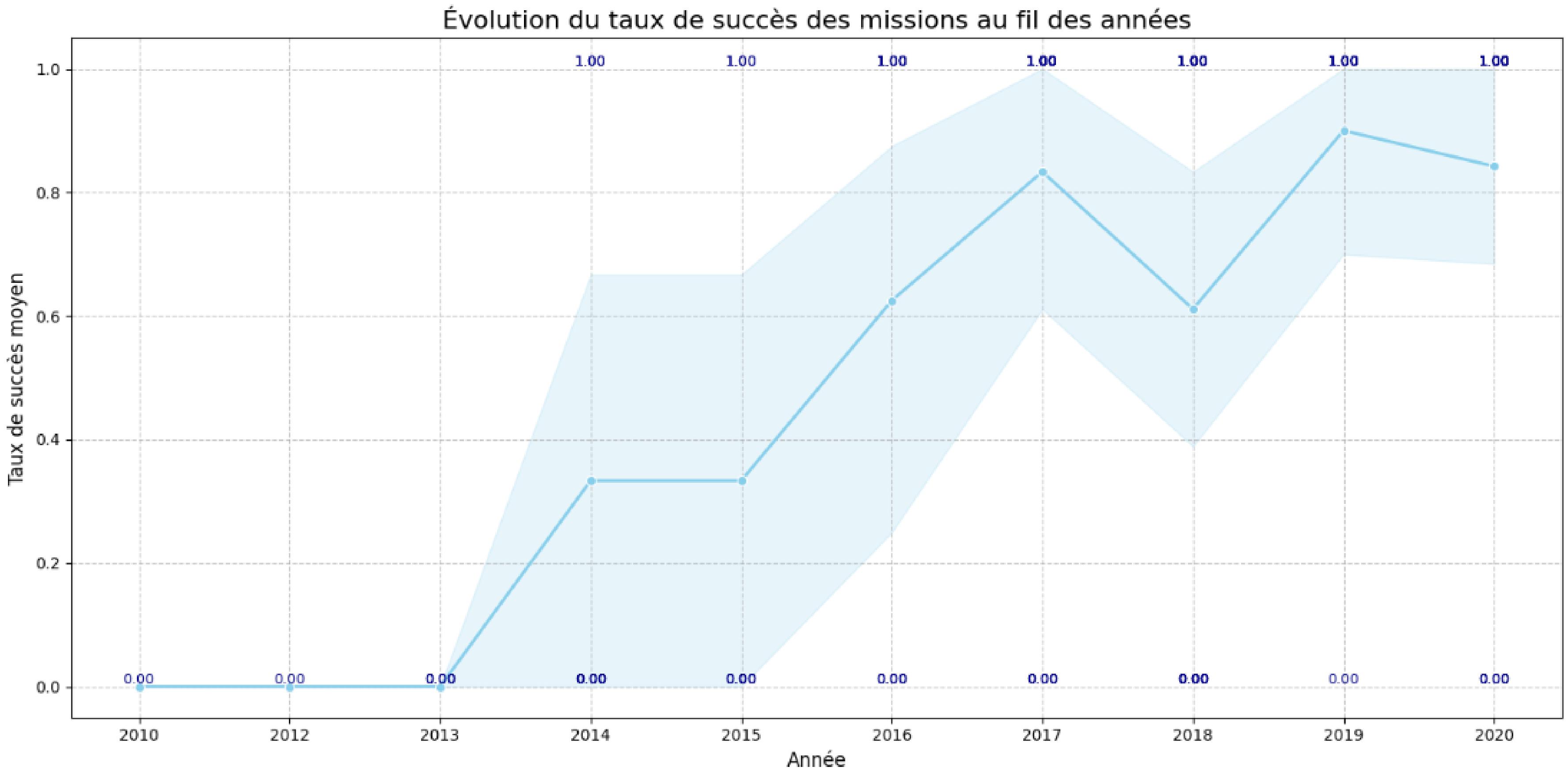
All details on the GitHub link (here)

File : edadataviz.ipynb

Purpose :

During the EDA process i visualized data using pandas / seaborn (for chart) in order to get a good general view and opinion of the data

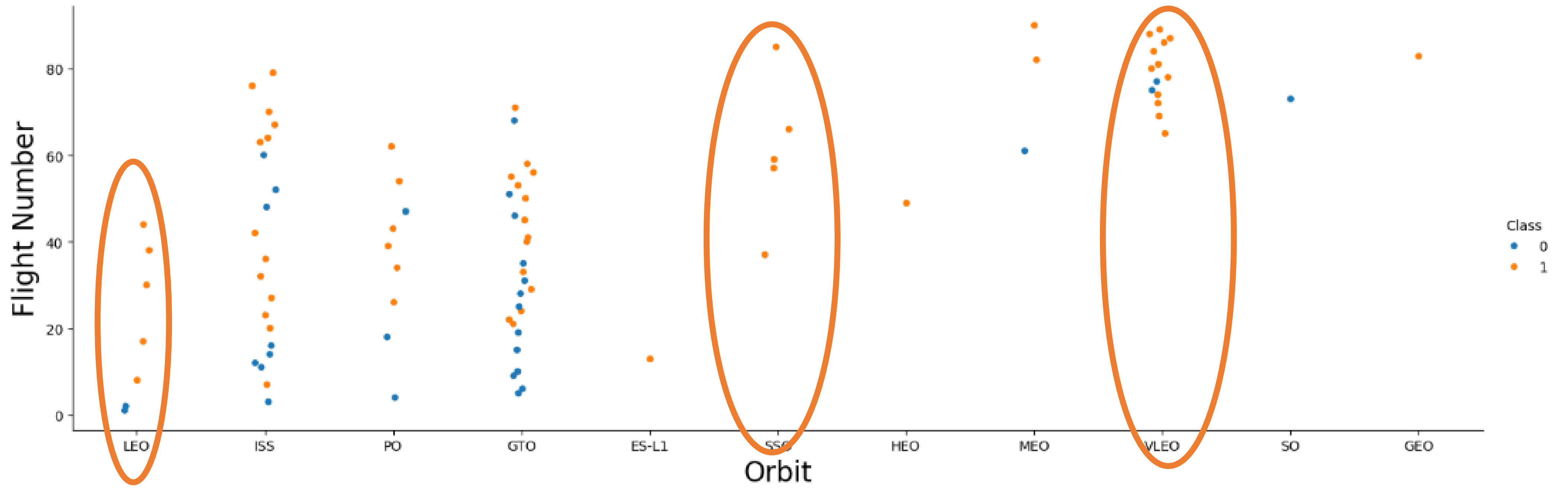




EDA with Data Visualization

Success evolution during the time

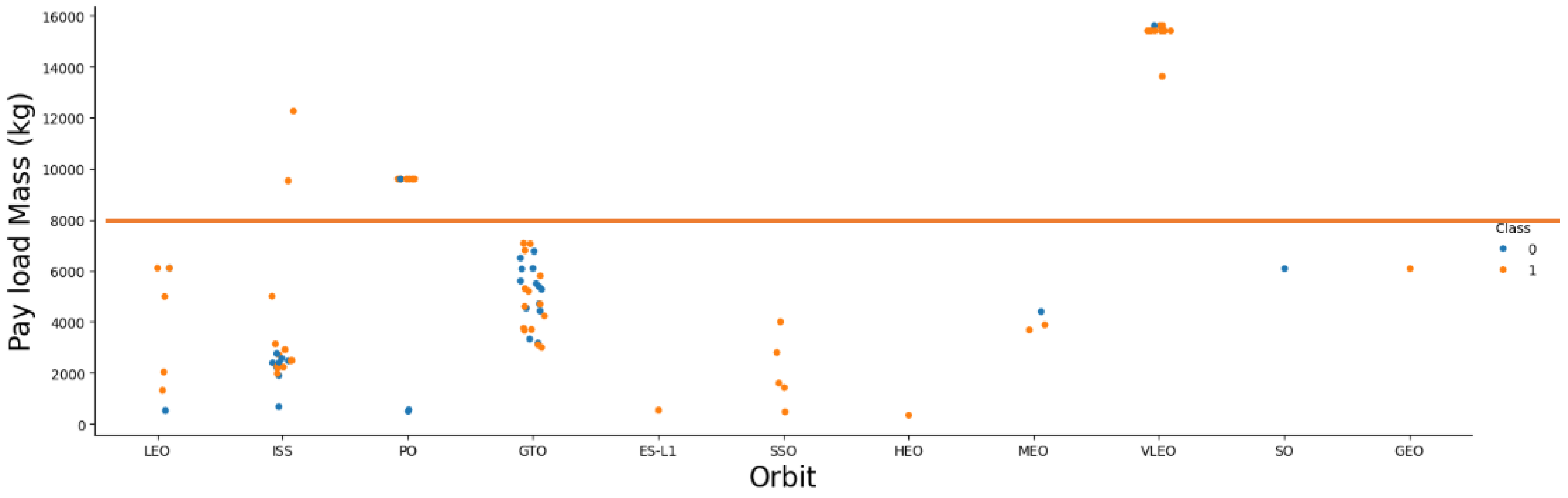
- Time & Experience showing a good success rate evolution



EDA with Data Visualization

<u>Best Success Orbit > 1 landing</u>

- LEO
- SSO
- VLEO



Best Success Orbit vs Payload Mass

- ISS / PO / VLEO for heavy payload
- ES-L1 / SSO / HEO for payload mass < 6000kg

EDA with Data Visualization

EDA with SQL

- Displayed names of Launch Sites
- Displayed 5 records where launch sites begin with the string 'CCA’
- Displayed the total payload mass carried by boosters launched by NASA (CRS)
- Displayed average payload mass carried by booster version F9 v1.1
- Listed the date when the first succesful landing outcome in ground pad was acheived
- Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listed the total number of successful and failure mission outcomes
- Listed all the booster_versions that have carried the maximum payload mass. Use a subquery.
- Listed the records which display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- Ranked the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

Details and results available (here)

File : jupyter-labs-eda-sql-coursera sqllite.ipynb

Build an Interactive Map with Folium

- After EDA, to get a global view on launch sites and success/fails i have indicated data on a map using Folium.

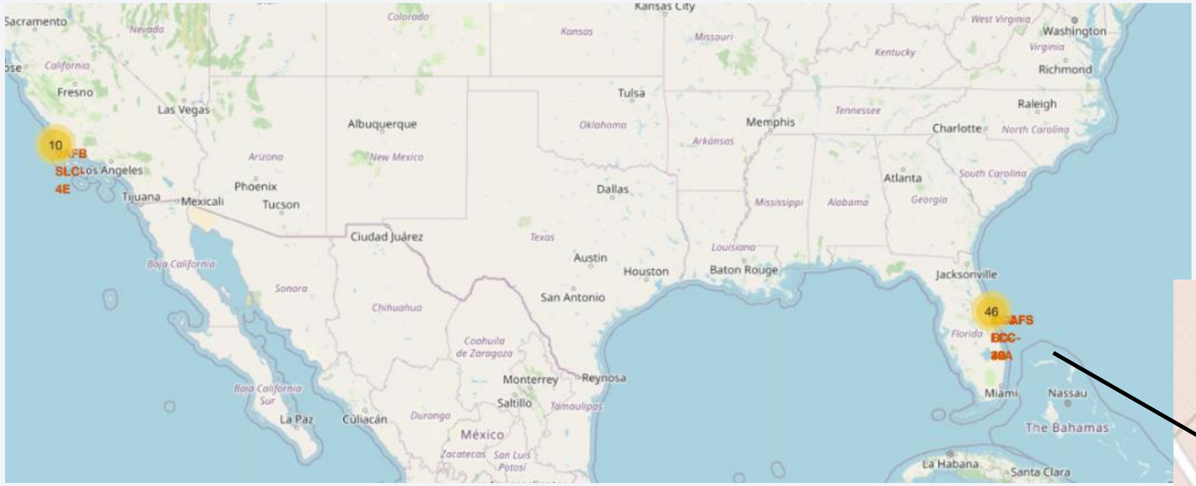
<u>**Notebook :**</u>

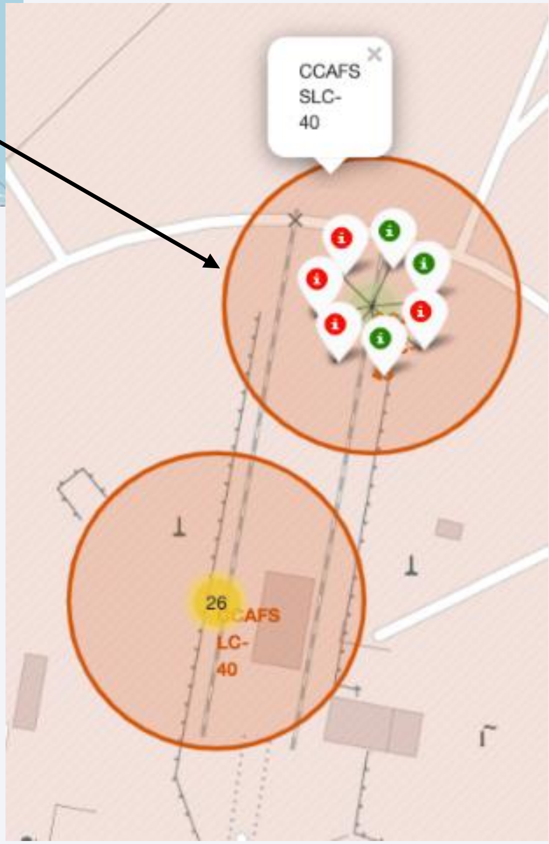
All details on the GitHub link ([here](#))

File :
<u>lab_jupyter_launch_site_location.ipynb</u>

Build an Interactive Map with Folium

- Marker_cluster creation to see the land status
  - failed = red marker
  - Sucess = green marker





By zooming on the map user can have a fast look at the location and mission status

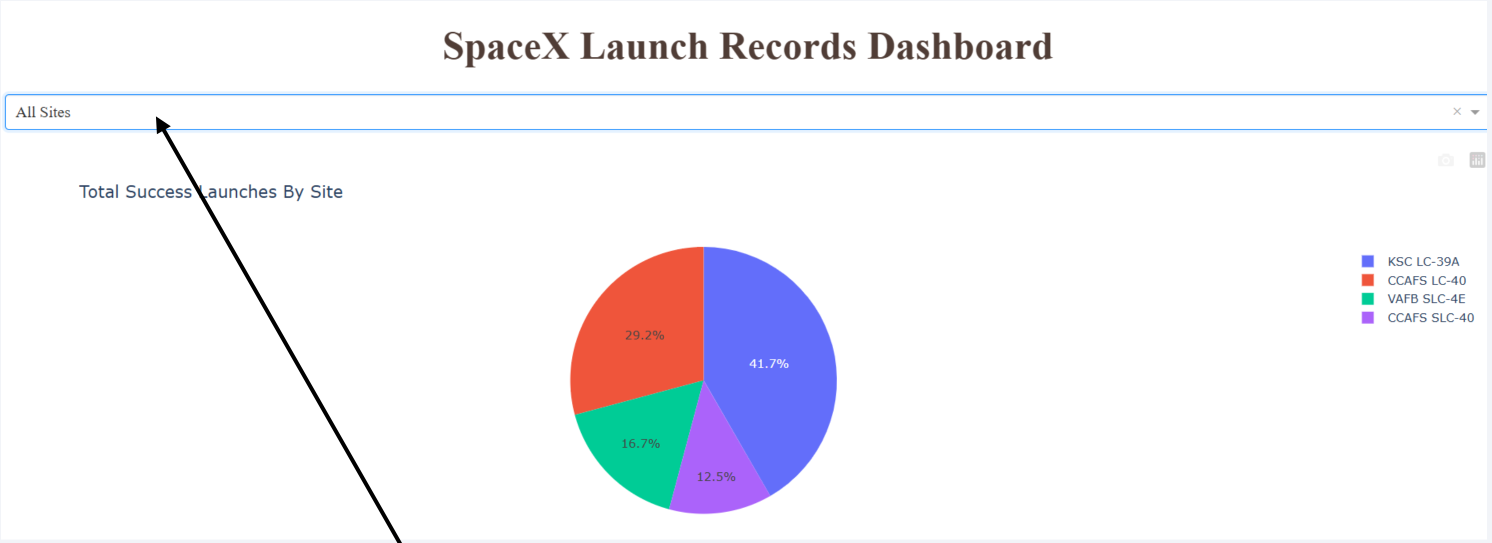
<u>Notebook :</u>

All details on the GitHub link (<u>here</u>)

File :
<u>lab_jupyter_launch_site_location.ipynb</u>

Build a Dashboard with Plotly Dash

- Using Ploty Dash i have created an interactive dashboard
- User can get a quick overview of the mission status



Select the Launch Site to get the details

Notebook :

Application Code details on the GitHub link ([here]())

File : Dash IDE.txt

Build a Dashboard with Plotly Dash

- As we saw previously, a key success factor is the payload mass
- With the dashboard, user can quickly see analysis result

Use the slicer to filter the payload range

<u>Notebook :</u>

Application Code details on the GitHub link (here)

File : <u>Dash IDE.txt</u>

Predictions

Now we got necessary historical information avec SpaceX Falcon 9 landing

- Keys factors are : **Launch Site / Orbit / Payload Mass**

Additionnaly, we knows that **time** & **experience** are going on the **favor of the project**

Based on those information, we will create machine learning models to predict the status of the futures missions

Predictive Analysis (Classification)

Source :

CSV File : **https dataset_part_2.csv + dataset_part_3.csv**

Created during Data Wrangling and EDA processes

Notebook :

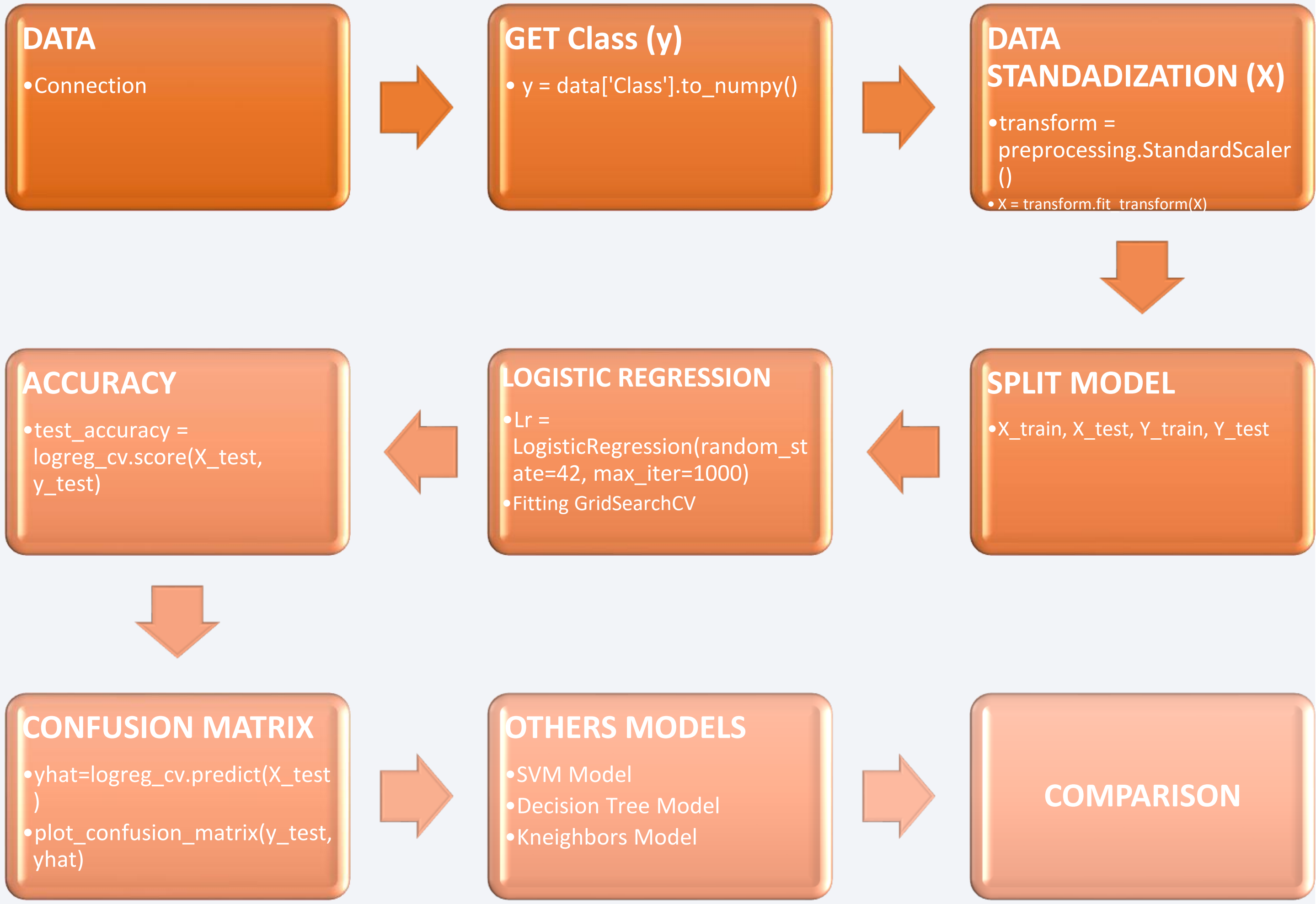
All details on the GitHub link (here)

File : SpaceX_Machine Learning Prediction_Part_5.ipynb

Purpose :

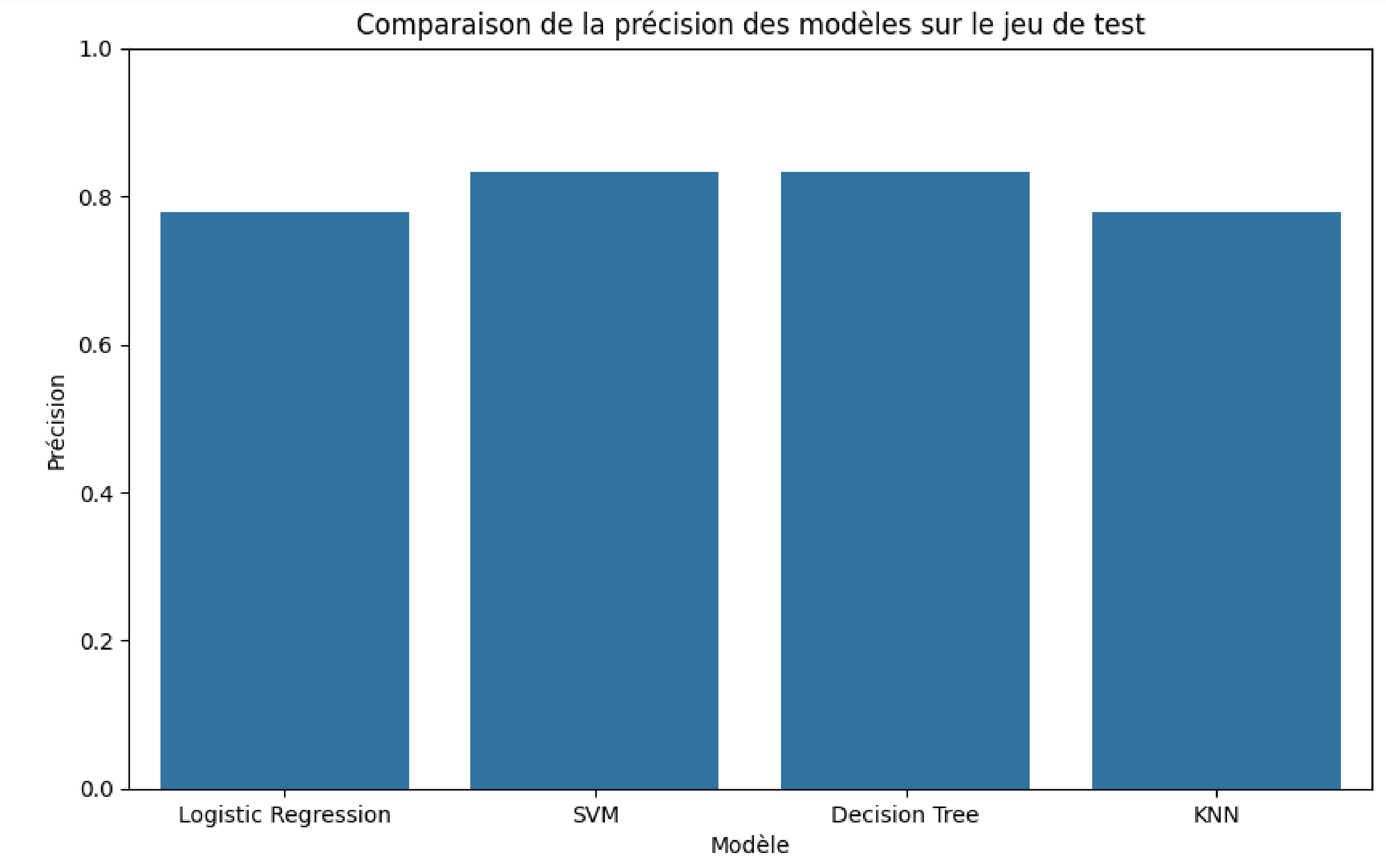
Create prediction Models and test it using confusion matrix.

Find the best prediction model



Predictive Analysis (Classification)

*SVM and Decision Tree Model got the best results and using Classification Report Evaluation Model, the **SVM Model get the best result***




```
Classification Report for Logistic Regression:
              precision    recall  f1-score   support

           0       1.00      0.33      0.50         6
           1       0.75      1.00      0.86        12

    accuracy                           0.78        18
   macro avg       0.88      0.67      0.68        18
weighted avg       0.83      0.78      0.74        18

Classification Report for SVM:
              precision    recall  f1-score   support

           0       1.00      0.50      0.67         6
           1       0.80      1.00      0.89        12

    accuracy                           0.83        18
   macro avg       0.90      0.75      0.78        18
weighted avg       0.87      0.83      0.81        18

Classification Report for Tree:
              precision    recall  f1-score   support

           0       0.80      0.67      0.73         6
           1       0.85      0.92      0.88        12

    accuracy                           0.83        18
   macro avg       0.82      0.79      0.80        18
weighted avg       0.83      0.83      0.83        18

Classification Report for KNN:
              precision    recall  f1-score   support

           0       1.00      0.33      0.50         6
           1       0.75      1.00      0.86        12

    accuracy                           0.78        18
   macro avg       0.88      0.67      0.68        18
weighted avg       0.83      0.78      0.74        18
```

Result

Back to our 3 questions :

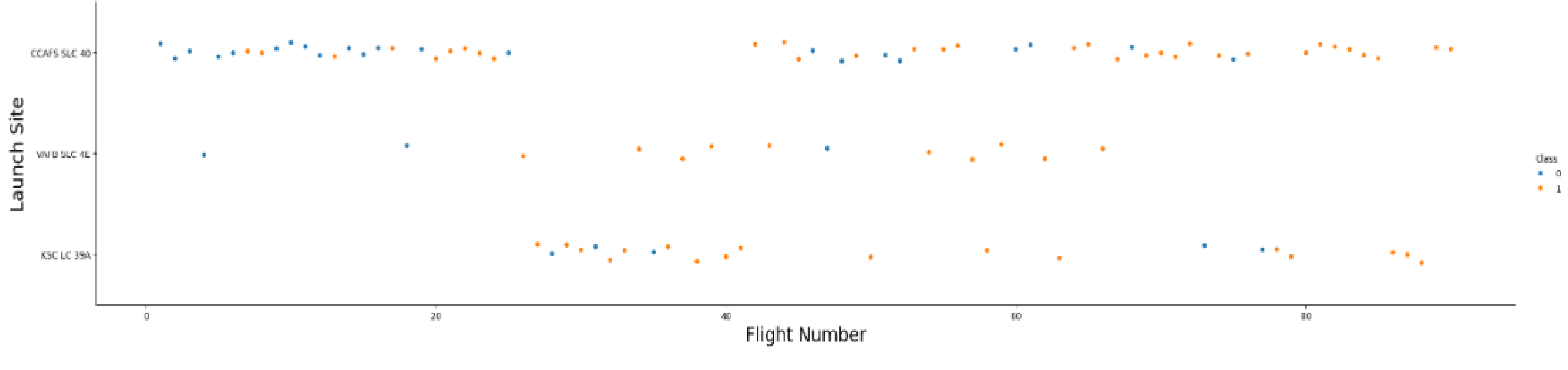
- What is the **best Orbit ?**
    - **LEO / SSO / VLEO are showing the best statistics landing success and ISS / PO / VLEO for heavy payload**
- What is the best **Pay Load mass range ?**
    - **Under 6000 Kg payload mass best orbits are ES-L1 / SSO / HEO**
- What is the best **Launch Site ?**
    - **KSC SC 39A got the best success rate**

Knowing this information, To predict the success of the next missions, we found then the Decision Tree and SVM Predictive Model got the best accurancy result and cross checked with Classification report **SVM Model** is the most adapted one

Section 2

Insights drawn from EDA

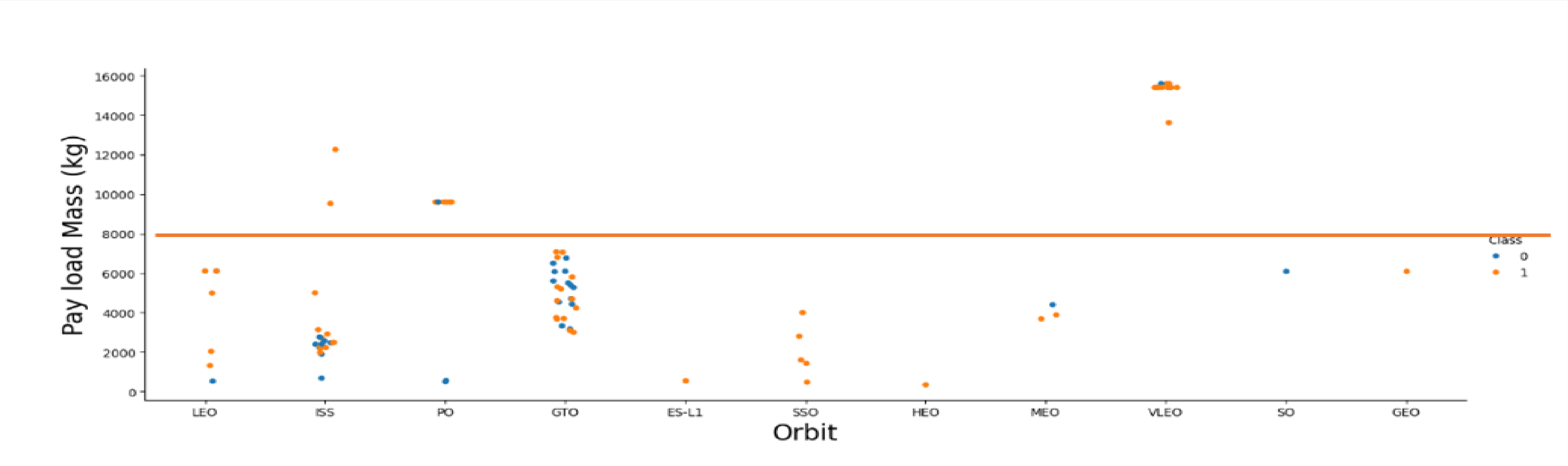
Flight Number vs. Launch Site



We can see a general increase of success with flight numbers.

After approx 30 flight numbers, the success rate increase.

Payload vs. Launch Site



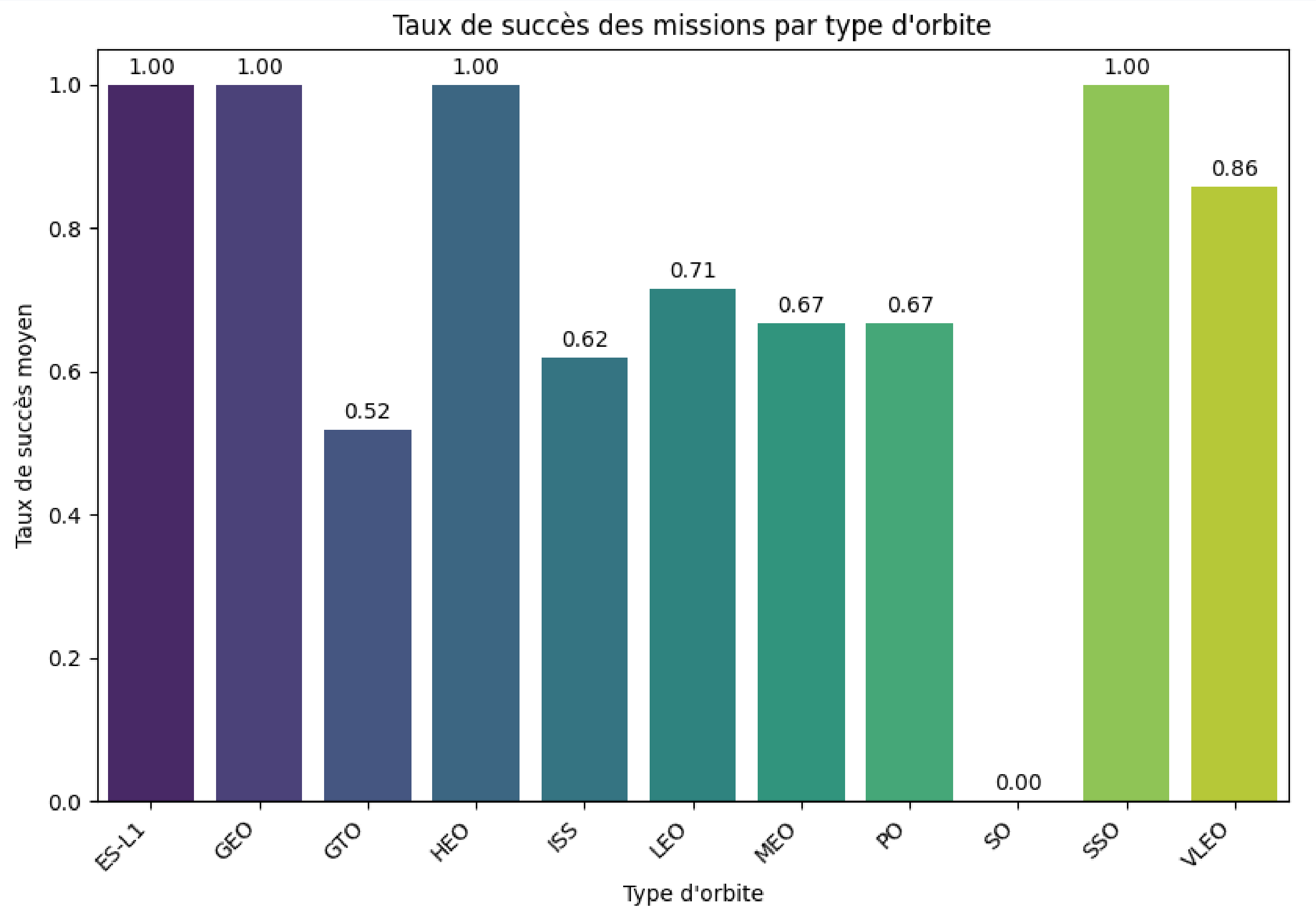
<u>Best Success Orbit vs Payload Mass</u>

- ISS / PO / VLEO for heavy payload
- ES-L1 / SSO / HEO for payload mass < 6000kg

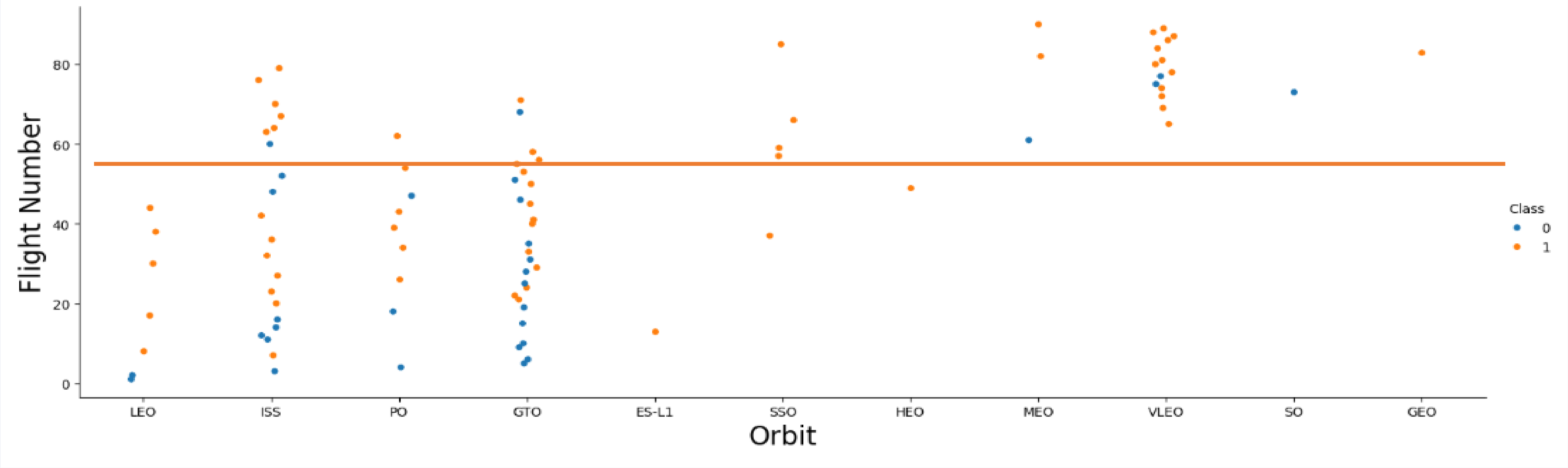
Success Rate vs. Orbit Type

4 orbit types got 100% success

It's a really interesting information for the next launches

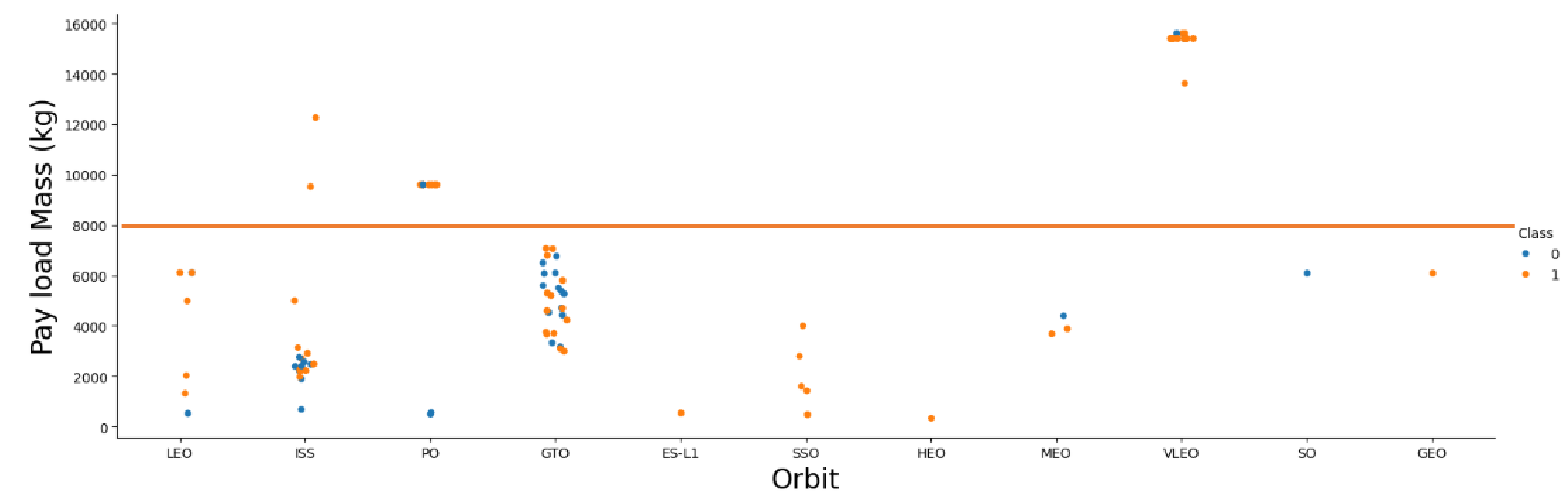


Flight Number vs. Orbit Type



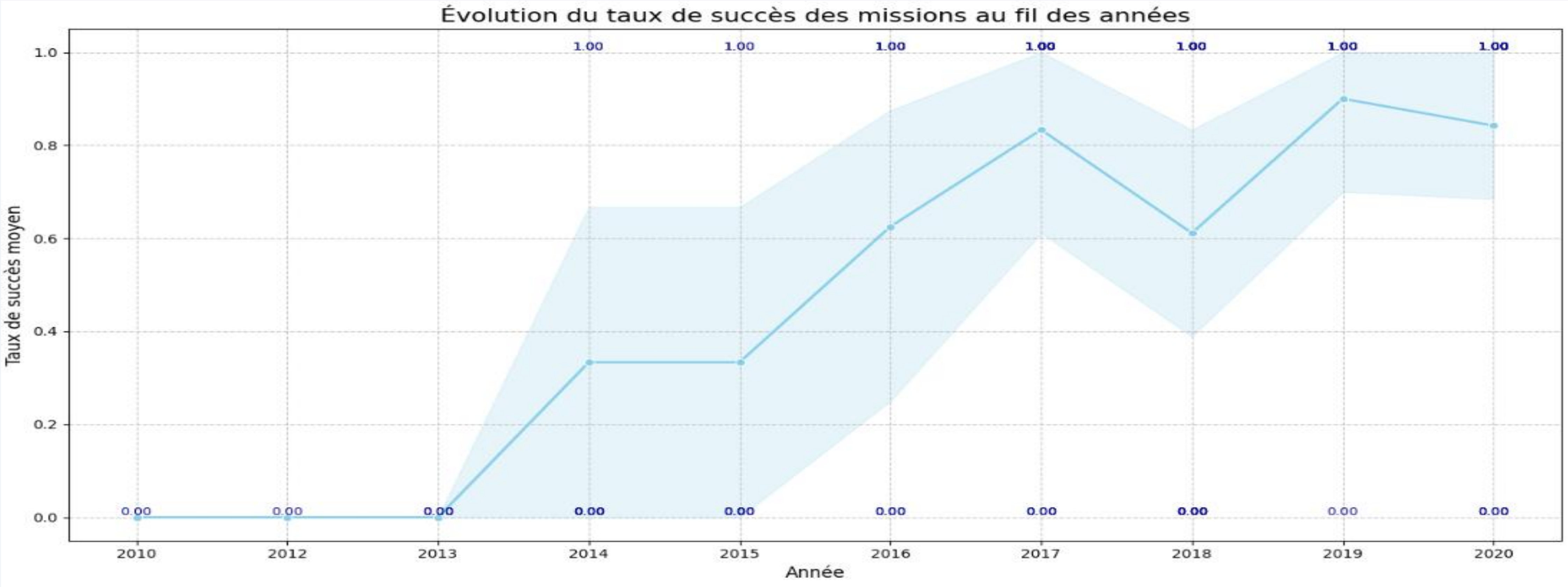
ISS / PO / GTO / SSO / VLEO are the choosen orbits from the lastest launches

Payload vs. Orbit Type



ISS / PO / VLEO are the choosen orbits for high Pay load Mass (> 8000kg)

Launch Success Yearly Trend



From 2015 to 2017, the success rate increased ; and from 2019 the success rate is above 80%

All Launch Site Names

From SQL query, we see 4 Launch Sites as below result :

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

Using SQL query, below the result of 5 launch sites begin with `CCA`

```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE '%CCA%'limit 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

From SQL query, we see the TOTAL payload mass carried by bosster from NASA :

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer LIKE 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

From SQL query, we see the AVERAGE payload mass carried by bosster F9 v1.1 :

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE booster_version LIKE '%F9 v1.1%';

 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

%sql SELECT MIN(Date), Booster_Version, Launch_Site, Payload, Orbit, Customer, Mission_Outcome, Landing_Outcome FROM SPACEXTABLE WHERE (Mission_Outcome LIKE 'Success') AND Landing_Outcome LIKE '%ground pad%';

```
yload, Orbit, Customer, Mission_Outcome, Landing_Outcome FROM SPACEXTABLE WHERE (Mission_Outcome LIKE 'Success') AND Landing_Outcome LIKE '%ground pad%';
```

```
 * sqlite:///my_data1.db
Done.
```

MIN(Date)	Booster_Version	Launch_Site	Payload	Orbit	Customer	Mission_Outcome	Landing_Outcome
2015-12-22	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	LEO	Orbcomm	Success	Success (ground pad)

First succes was done on the 2015-12-22 from CCAFS LC-40 using the orbit LEO

Successful Drone Ship Landing with Payload between 4000 and 6000

%sql SELECT DISTINCT(Booster_Version), Launch_Site, Payload, PAYLOAD_MASS__KG_, Orbit, Customer, Landing_Outcome FROM SPACEXTABLE WHERE (Landing_Outcome LIKE '%Success (drone ship)%') AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)

```
%sql SELECT DISTINCT(Booster_Version), Launch_Site, Payload, PAYLOAD_MASS__KG_, Orbit, Customer, Landing_Outcome FROM SPACEXTABLE WHERE (Landing_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Landing_Outcome
F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success (drone ship)
F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success (drone ship)
F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success (drone ship)
F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success (drone ship)

From SQL query, we see 4 Drone Ship Landing with Payload > 4000 and < 6000.

Additionnaly, we can see from 2 Launch Sites (CCAFS and KSC with the Orbit GTO

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT COUNT(Mission_Outcome),Mission_Outcome  FROM SPACEXTABLE GROUP BY Mission_Outcome ;
```

```
 * sqlite:///my_data1.db
Done.
```

COUNT(Mission_Outcome)	Mission_Outcome
1	Failure (in flight)
98	Success
1	Success
1	Success (payload status unclear)

From SQL query, we see that on 101 Mission Outcomes, there is only 1 failure

Boosters Carried Maximum Payload

From SQL query, we see the MAX Payload carried by boosters

```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

* sqlite:///my_data1.db
Done.

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

From SQL query, we see the list of failed outcomes in drone ship

Done.

	Date	YEAR	Month	Booster_Version	Launch_Site	Landing_Outcome
	2015-01-10	2015	January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
	2015-04-14	2015	April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Remark : due to the regionnal parameters setup on my computer, i had to convert date in the SQL Query as below in order to show the month name

```
%sql SELECT Date, substr(Date, 0, 5) AS YEAR, CASE substr(Date, 6, 2) WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' WHEN '04' THEN 'April' WHEN '05' THEN 'May' WHEN '06' THEN 'June' WHEN '07' THEN 'July' WHEN '08' THEN 'August' WHEN '09' THEN 'September' WHEN '10' THEN 'October' WHEN '11' THEN 'November' WHEN '12' THEN 'December' ELSE '' END AS Month, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE WHERE (Landing_Outcome LIKE '%Failure (drone ship)%') AND YEAR LIKE '2015';
```

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

List and details of Outcomes between 2010-06-04 and 2017-03-20

SQL Query Detail :

%sql SELECT * FROM SPACEXTABLE WHERE (Date BETWEEN '2010-06-04' AND '2017-03-20') AND (Landing_Outcome LIKE '%Failure (drone ship)%') OR (Landing_Outcome LIKE '%Success (ground pad)%');

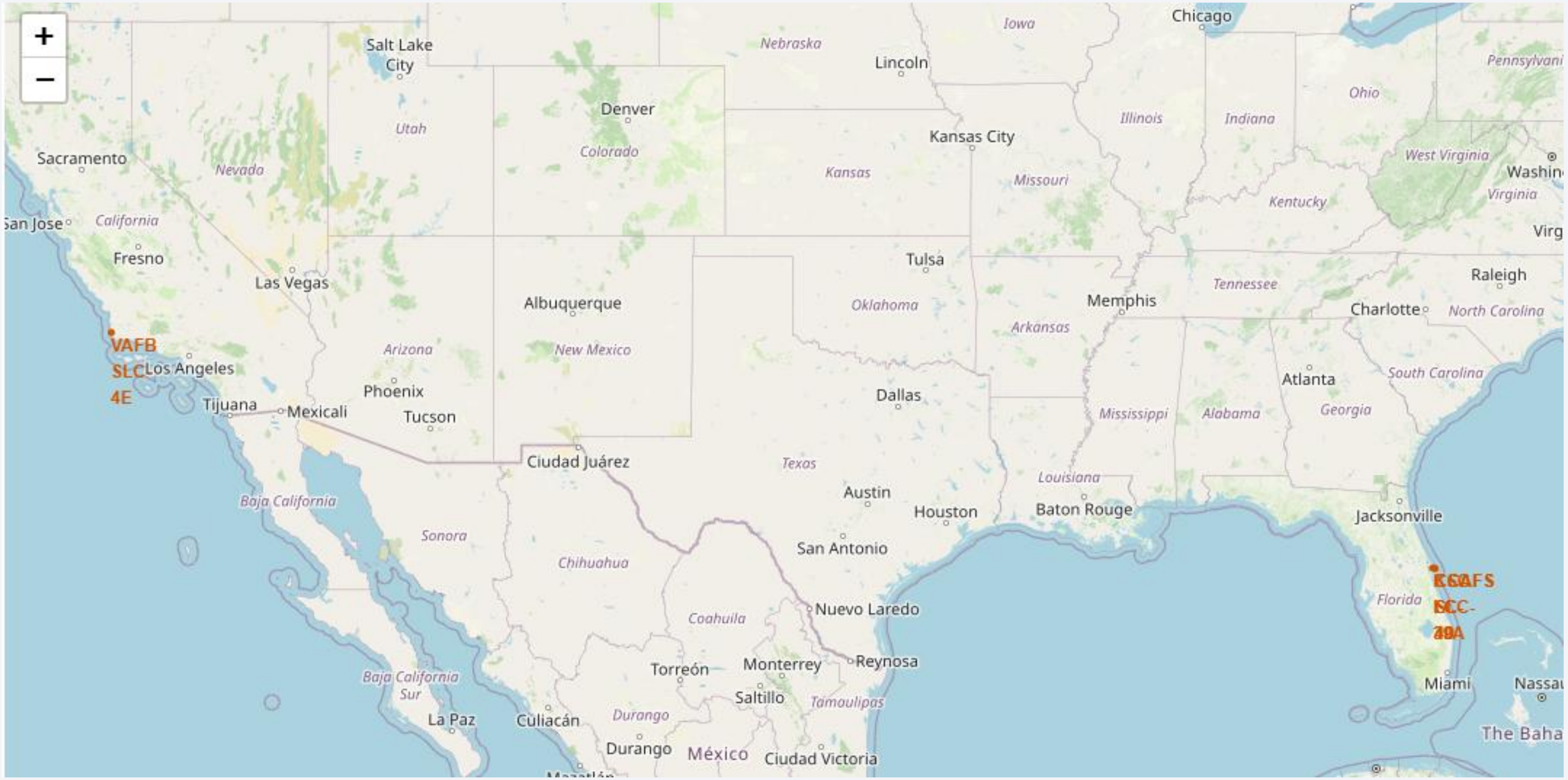
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2015-01-10	9:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)
2016-01-17	18:42:00	F9 v1.1 B1017	VAFB SLC-4E	Jason-3	553	LEO	NASA (LSP) NOAA CNES	Success	Failure (drone ship)
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	Success	Failure (drone ship)
2016-06-15	14:29:00	F9 FT B1024	CCAFS LC-40	ABS-2A Eutelsat 117 West B	3600	GTO	ABS Eutelsat	Success	Failure (drone ship)
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-06-03	21:07:00	F9 FT B1035.1	KSC LC-39A	SpaceX CRS-11	2708	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-08-14	16:31:00	F9 B4 B1039.1	KSC LC-39A	SpaceX CRS-12	3310	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-09-07	14:00:00	F9 B4 B1040.1	KSC LC-39A	Boeing X-37B OTV-5	4990	LEO	U.S. Air Force	Success	Success (ground pad)
2017-12-15	15:36:00	F9 FT B1035.2	CCAFS SLC-40	SpaceX CRS-13	2205	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2018-01-08	1:00:00	F9 B4 B1043.1	CCAFS SLC-40	Zuma	5000	LEO	Northrop Grumman	Success (payload status unclear)	Success (ground pad)

Section 3

Launch Sites Proximities Analysis

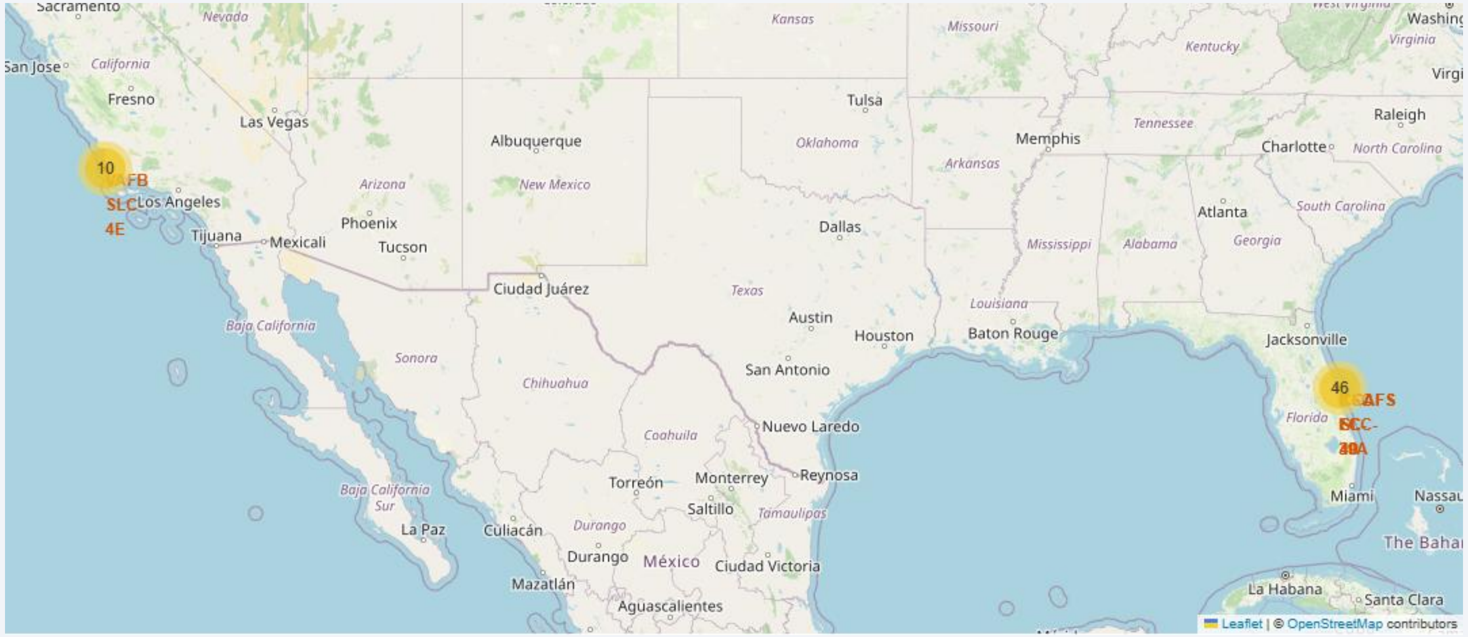
Launch Sites

Using Folium and marker i labeled the launch sites on a map



Launch Sites – Marker clusters

I added marker clusters on the map to be able to see on the map the mission status :



Red = Failure

Green = Succes

Launch Sites – Distances with its proximities

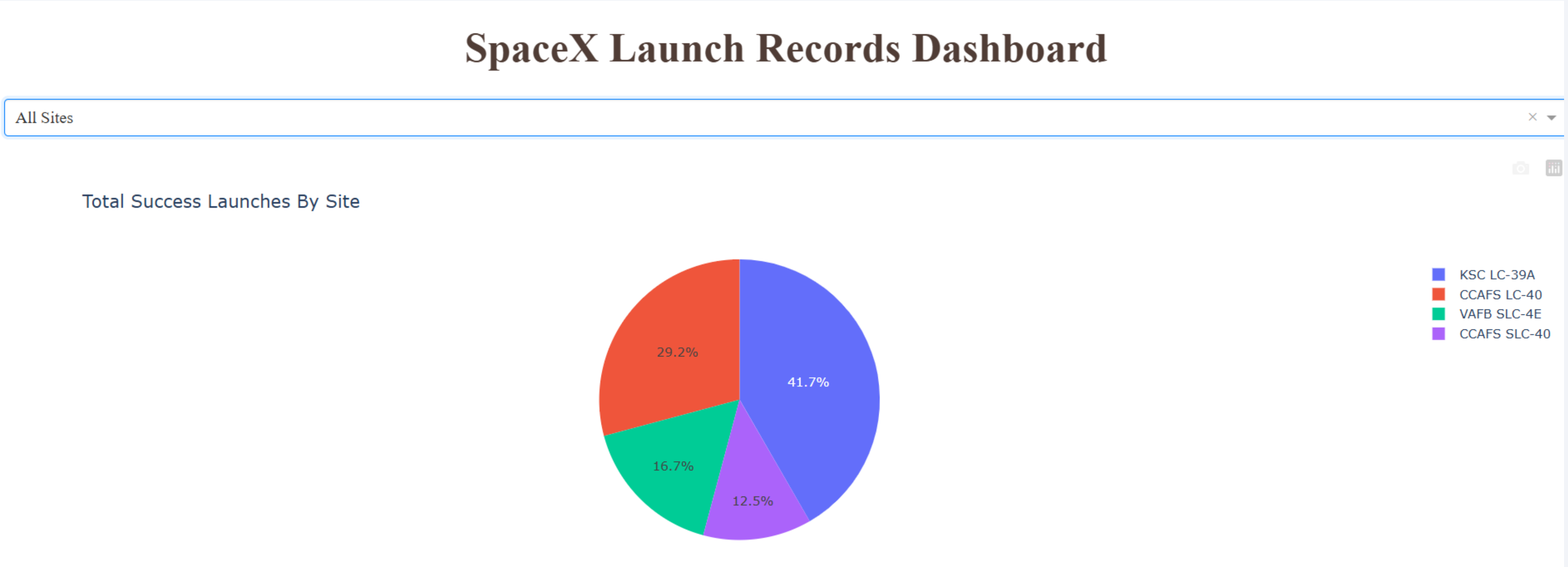
Using folium.PolyLine we can see the distance between launch site and its proximities is around 0,90km

Section 4

Build a Dashboard with Plotly Dash

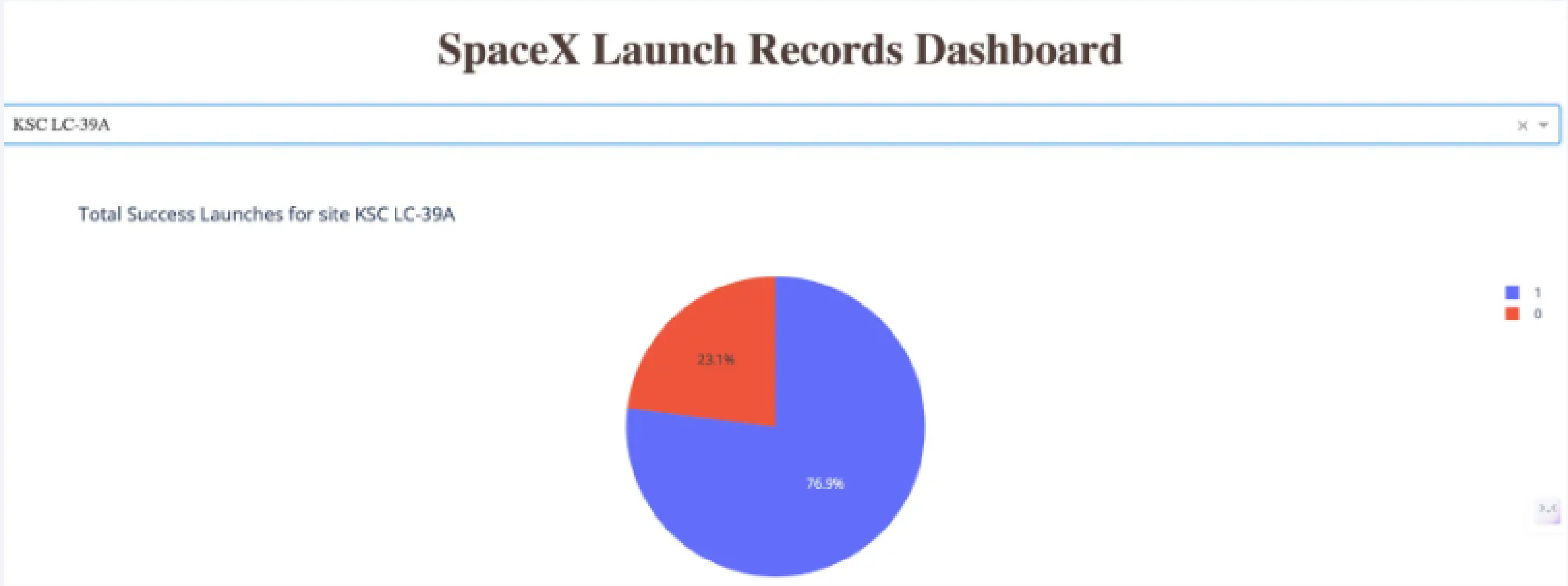
Plotly Dash – Success by Launch Sites

I have created an interactive dashboard with Dash IDE. It result KSC launch site got the highest success rate with 41,7%



Plotly Dash – Zoom on KSC Launch Site

We can see from KSC Launch Site, the success rate is 76,9%



Plotly Dash – Payload vs launch Outcome

The booster version on the category FT got the best success rate.

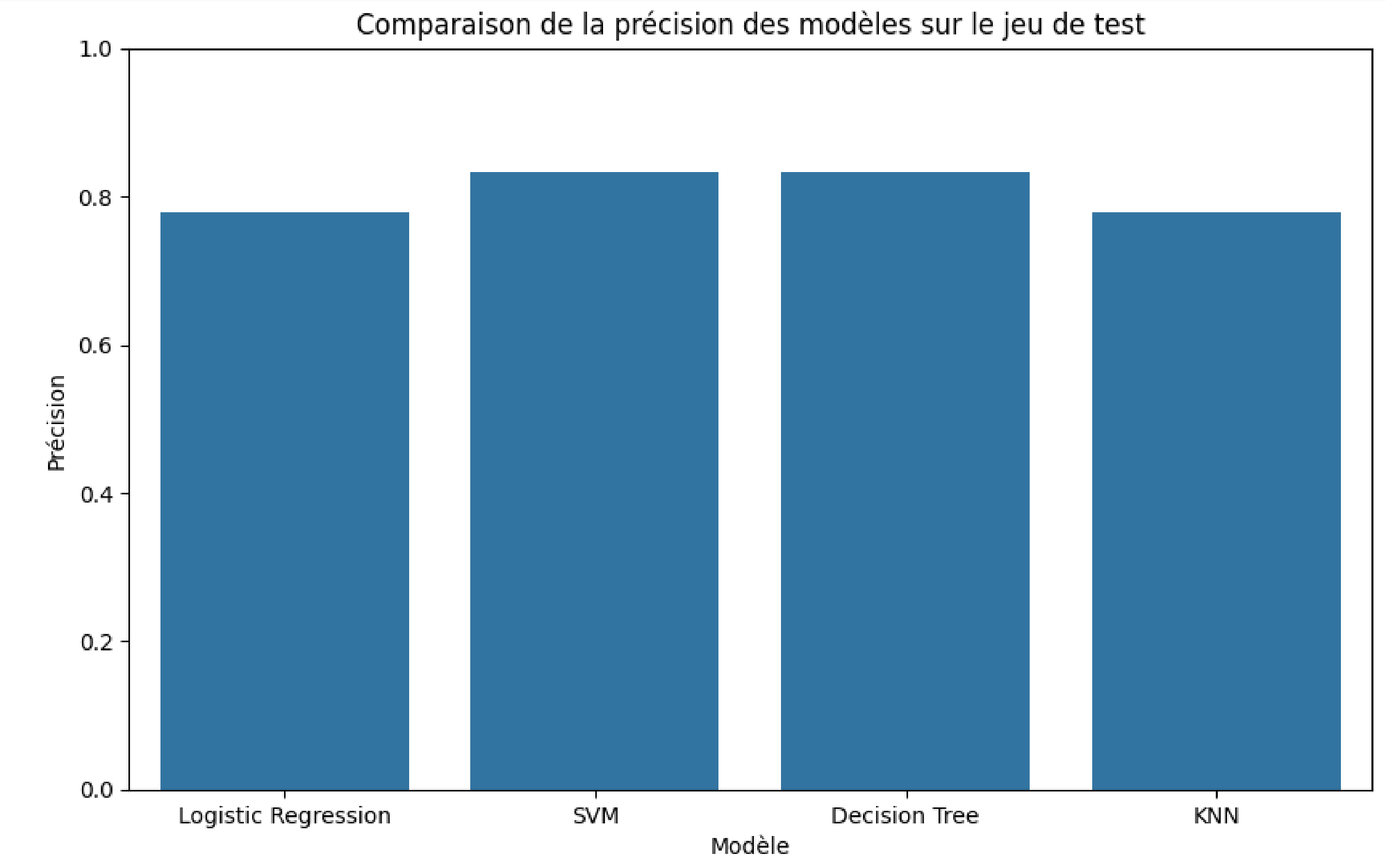
The most successful payload mass is between 2000kg and 4000kg.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

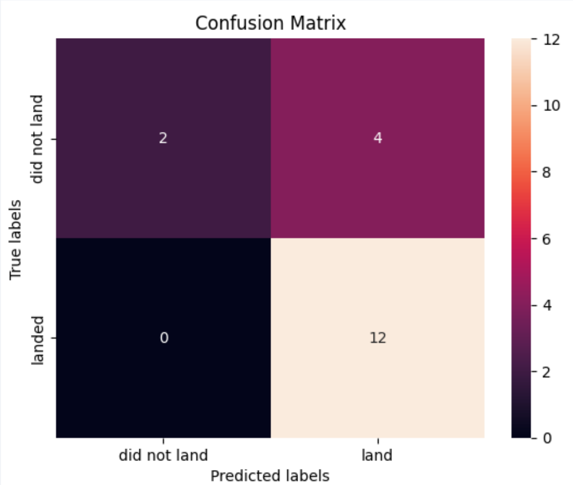
*SVM and Decision Tree Model got the best results and using Classification Report Evaluation Model, the **SVM Model get the best result***



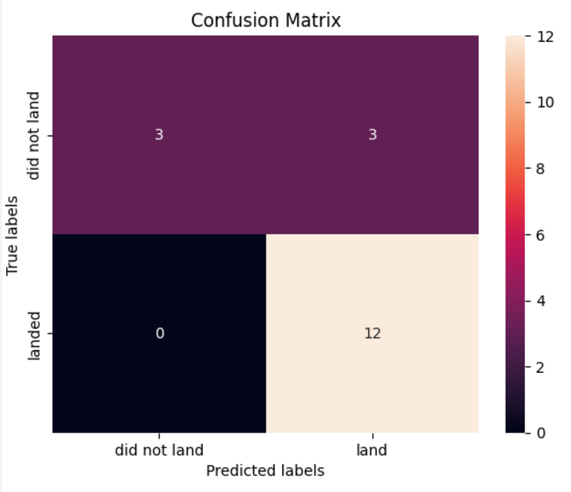

```
Classification Report for Logistic Regression:
              precision    recall  f1-score   support

           0       1.00      0.33      0.50         6
           1       0.75      1.00      0.86        12

    accuracy                           0.78        18
   macro avg       0.88      0.67      0.68        18
weighted avg       0.83      0.78      0.74        18

Classification Report for SVM:
              precision    recall  f1-score   support

           0       1.00      0.50      0.67         6
           1       0.80      1.00      0.89        12

    accuracy                           0.83        18
   macro avg       0.90      0.75      0.78        18
weighted avg       0.87      0.83      0.81        18

Classification Report for Tree:
              precision    recall  f1-score   support

           0       0.80      0.67      0.73         6
           1       0.85      0.92      0.88        12

    accuracy                           0.83        18
   macro avg       0.82      0.79      0.80        18
weighted avg       0.83      0.83      0.83        18

Classification Report for KNN:
              precision    recall  f1-score   support

           0       1.00      0.33      0.50         6
           1       0.75      1.00      0.86        12

    accuracy                           0.78        18
   macro avg       0.88      0.67      0.68        18
weighted avg       0.83      0.78      0.74        18
```

Confusion Matrix

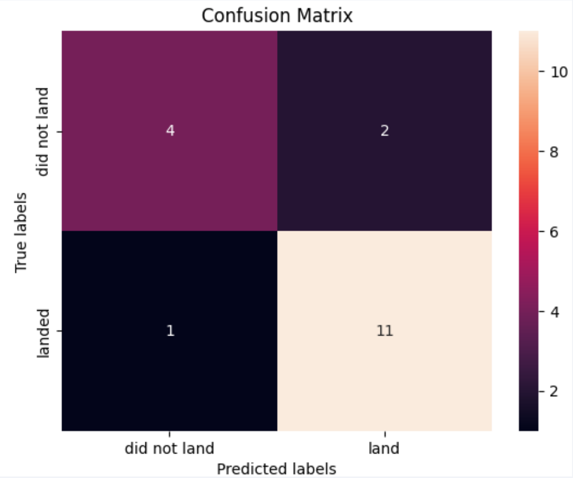
Logistic Regression



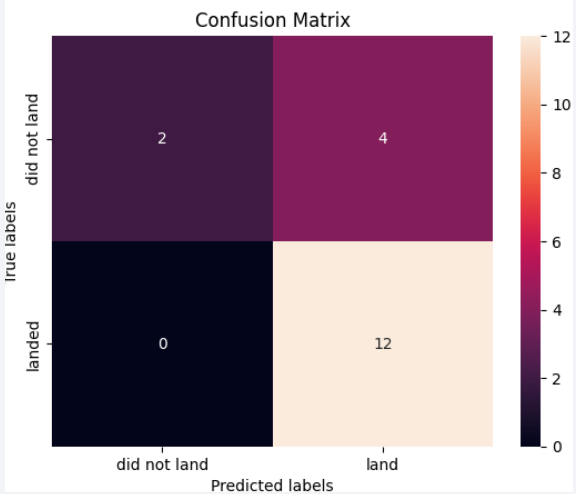
Support Vector Machine



Decision Tree



K-Nearest Neighbors



	Train_Accuracy	Test_Accuracy
Logistic Regression	0.862500	0.777778
SVM	0.850000	0.833333
Decision Tree	0.889286	0.833333
KNN	0.878571	0.777778

On the training dataframe, the Decision Tree is giving the best result but then on the testing model, SVM and Decision Tree gives 83% accuracy

Conclusions

- LR and KNN shows a result of 4 false positive vs 3 using SVM and 2 for Decision Tree.
- Based on classification Report, SVM model with Precision gives 90% accuracy
- It can be interesting to test the both model SVM and Decision Tree with more data to define which one is working the best.

Thank you!